International Journal of Computer Science Engineering
and Information Technology Research (IJCSEITR)
ISSN (P): 2249-6831; ISSN(E): 85-92
Vol. 3, Issue 5, Dec 2013, 223-232
© TJPRC Pvt. Ltd.

DEVELOP OBJECT BASED DEBUGGING TOOL TECHNIQUE FOR IMPROVING THE PERFORMANCE OF OBJECT ORIENTED SOFTWARES

D. M. THAKORE[1] & TANVEER S BEG[2]

[1]Professor, Bharati Vidyapeeth Deemed University College of Engineering, BVDUCOE, Pune, Maharashtra, India

[2]Reaserch Scholar, Computer Student, Bharati Vidyapeeth Deemed University College of Engineering, BVDUCOE, Pune, Maharashtra, India

ABSTRACT

Various functionalities and debugging techniques during software development may end with some blunder mistakes in later stages of software development, due to which intended and projected functionality of the software is difficult to achieve. In the process of Software Development and evolution, Developer has to answer multiple questions about how the code or software behaves at runtime. The traditional or classical debugger while debugging gives developer bunch of breakpoints in the source code. This is an vague and not consistent stage which is difficult to be used by the developer for development. Some of debugging tools are helpful for understanding the problems, as acknowledged from conventional tools that the difficulty of object oriented system expands, debugging becomes by a long way difficult. Developer needs a devoted user interface for such type of these operations related to objects and other; this need is pleased by facilitating a user interface for the programmer.

Object based debugging tool looks ahead to analyse the relationship in between the objects at some point in the runtime. There exists therefore conceptual space between the interface offered by the debugger and the need of the developer, hence to beat this drawback or problem; there is a need for object based debugger and useful interface for it. In this paper, presented a prototype of such debugger and there results and discussions in context of programmer point of view, so that programmer can get clear idea about intended functionality of their developing prototype or software.

KEYWORDS: Software Programming, Objects, Debugging, UI, Errors

INTRODUCTION

Traditional or Classical Debugger always used to concentrate on the execution of stack. In this scenario programmer recognises parts of source code of interest and sets breakpoints accordingly. These breakpoints are set of purely with respect to static abstraction and not respect to particular object of the running system. This Object based debugging as an alternative approach to interacting with a running software system. Mainly Focusing on objects as the key point, natural debugging operations can be defined to answer developer questions related to runtime behaviour. Here, presented scenario of an object based debugger. How it offers more effective support for many typical developer tasks than a traditional or classical debugger.

Traditional or Classical debuggers are not always up to the task, since they bind to provide access to information that is still in the run-time stack. In some cases, the information needed to track down these difficult bugs content; how an object reference got here, the previous values of object's fields. For this reason it is helpful to have previous object states and object reference flow information at hand during debugging.

RELATED WORKS

Debugging Tools

At the present day's plenty of debugger tools are at hand. The capabilities of some of the existing debuggers some of them explained and discussed. Such as jBixbe, DBG | PHP, Jswat, Backstop tool, WPOL, CMeRun, CnC and OOCD. Bixbe apply this in Java applications on the theoretical level at which they are designed it possible to find simple bugs as well as weaknesses or insufficiencies in application given. It shows the details of application selecting particular classes, objects, with their relationships. jBixbe provides a new feature of debugging complex Java applications by allowing their structure and functioning on the conceptual stage of the UML. This debugger also explains object-oriented concepts and provides source code debugging and breakpoints. But there are few limitations of these debuggers. Some researches felt that it is complex when debug a large application because jBixbe is made for high level object-oriented Java debugger. That's why it is very difficult for entry level learner to study or understand the error because the jBixbe can't locate the error. This debugger do not have pop-up window to tell a user what to do if errors occurs. The next example is a DBG | PHP–Debugger DBG (NuSphere, 2009) which is an open source debugger profiler for PHP programming language. PHP Debugger which is best tool for helping the bugs fast and eliminates them from the PHP programs. It supports not only a GUI interface but also command-line interface. DBG acts as full-featured PHP debugger, an interactive tool, helps debugging PHP scripts. Different types of debuggers are consisting of their independent and ideal functionalities but typical some debuggers where consisting of following architecture Figure 1 Architecture is shown in following diagram [14].

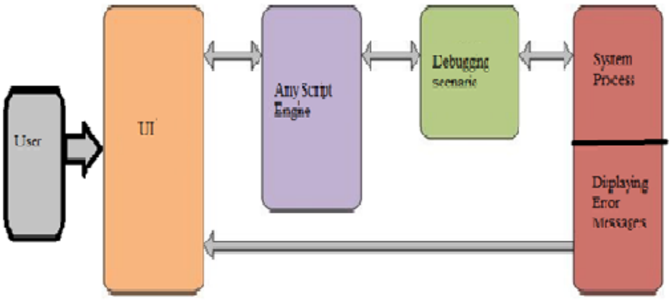


Figure 1: Typical Architecture Debugger

It acts on development web server, allows debug your scripts locally or remotely, from an IDE or console. PHP Debugger provides a powerful as well as easy way to simplify PHP debugging because it allows complete visibility as well as control over the execution of PHP scripts. This debugger also doesn't require that you make any changes to your PHP code. PHP Debugger can be debugging PHP applications on eighteen different platforms locally and remotely. Benifits for the debugging of nested calls multiple debug sessions as well as debugging of PHP CLI scripts set PhpED apart from other PHP IDE's. The advantages of DBG| PHP Debugger are allow user step by step through the execution of a PHP scripts, line-by-line as well as user friendly GUI. Even though with benefit of that DBG | PHP-Debugger can give, but there are some researches implement need Object-oriented Java Debugger which cannot support by this debugger.

Another drawback is PHP is an old script or many of organization have change to JSP in web development. The next example is a Jswat debugger (Swat, 2009). Which is a standalone and a graphical Java debugger, made to use the Java Platform Debugger Architecture. The Jswat features, breakpoints with conditionals as well as monitors actions, source code display, graphical display showing all threads, stack frames, visible variables as well as loaded classes, command interface for advanced features or Java-like expression evaluation including method invocation.

The benifits of Jswat debugger which are simple and user friendly GUI. This debugger is best for analyzing applications (maintenance) or display object relationships (structure diagrams). Even with features that Jswat debugger can give, it also have some drawbacks such as do not have pop-up window to tell a user what to do in step by step through the execution of a PHP scripts or do not show as a line by line but it just only pin-pointed on the specific error. Backstop tool (Murphy *et al.*, 2004), select the common runtime error in Java applications or do not identify the logic error. This tool devoloped for programmers studying Java at the entry level as well as it provides more user-friendly error messages when an uncaught runtime error (exception) occurs. It also gives the debugging support by users to watch the execution of the program and the changes to the values of variables. Same way with the educational tool Expresso proposed by Hristova *et al.* (2003) select or identifies the common error in Java programming and generates error messages which provide suggestions on how to fix the code. Those existing tools have been designed to identify logic errors faster but not give any suggestions to resolve them.

There also has been investigation of debugging techniques among the entry level programmers. The tool that presented in this research can be used to do debugging techniques among novice stable or not fragile. CAP (Schorsch, 1995), developed to aid entry level programmers in a user-friendly fashion by getting all syntax logic as well as style errors that they make in Pascal program. Which also supply information to the beginners about the error, the reason error has been occur as well as give the solution to fix the problem. According to Ebrahimi and Schweikert (2006), novices may not detect negative interactions between section and block of code. For example the code to perform the output which is correct but in the wrong place in the program. So WPOL (Ebrahimi and Schweikert, 2006) designed to facing the problem. WPOL being designed to incorporate the Plan-Object-Paradigm, Web as well as assessment with focus on plan integration. WPOL is plan object-oriented teaches novices programming by plan management as to how they are integrated as well as bridges the gap between object and functions. A plan that used in WPOL is an abstraction of a concept, requirement, object and programming code. The plan is used for structured knowledge representation in natural language processing. Another debugger is Check‘n’ Crash (CnC) introduced by Csallner and Smaragdakis (2005) is an automatic error detection which combines the static checking as well as automatic test generation to get the best of both function in order to detecting errors. The CnC tool combines the advantages of ESC/Java static checker and JCrasher tool automating testing tool. This tool, it consist of taking the abstract error conditions using theorem proving techniques by a ESC/Java static checker and deriving the specific error conditions using a constraint solver which then produce concrete test cases that are executed to determine whether an error truly exists by JCrasher tool.

It can show for beginners what happens when the code is executed. Visual debugger for Java programs (JVD) introduced by Rafieymehr and McKeever (2007) is developed using the graphical animation as well as runtime state retention to display program state during execution. These functions to detect runtime errors by determine which classes have main methods as well as ask user to choose one and display the code with highlight showing current line. Giving compiler type error messages is challenging for novice programmers (Hartmam *et al.*, 2010). HelpMeOut by Hartmam *et al.* (2010) is a tool that aids novices with the debugging of compiler error messages by giving successful solution to the errors that other programmers have got already [14]. Many tools have been designed in debugging area to find bugs in software but some of available technique are difficult to use and not benefits in finding real bugs. In scenario to study in depth of programming processes there have two types of vital thing. In the beginning they control the knowledge structures that programmers possess if they wish to measure the effects of factors that influence programmer performance. One more is, the researchers should understand the knowledge structures that beginner programmers have (Vessey, 1985).

This research has concentrated on the problems that occur in debugging process to object-oriented programming among the novice programmers. Next discuss the ways that can be used to improve the learning of object oriented programming. Data Debugging in this given is an approach for locating potential data errors. When it is impossible to know whether data are full of errors or not, data debugging does the next best thing: locating data that has an unusual impact on the computation. Intuitively, data that has vital impact on the last result very important, By contrast, wrong data whose presence has no particularly unusual effect on the final result does not merit special attention. Data debugging get together data dependence analysis and statistical analysis to find as well as rank data based on the unusualness of its impact on the results of a computation.

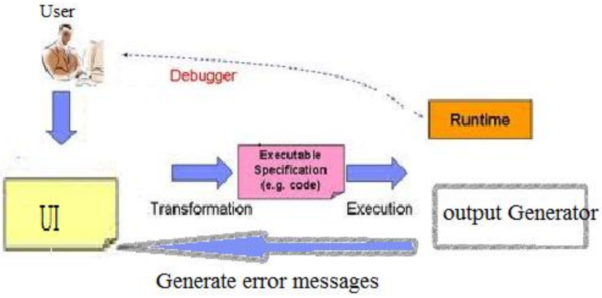


Figure 2: Proposed Architecture of System

As shown in Figure 2 Proposed architecture of system. The source code when debug using object based debugging tool, particular object required by developer is searched and made available to developer. Developer further acting on object do the specified operation by using user interface concentrated on objects. The source code when debug using object based debugging tool, particular object required by developer is searched and made available to developer. Developer further acting on object do the specified operation by using user interface concentrated on objects. The code file taken into proposed tool, then code parsing done for all particular objects [15]. After going through execution and isolates the points needed by developer needs. The parser extracted all objects from provided code file then supplied or given to execution module. This parser also converted it into intermediate forms which give response to object related errors or bugs. In code generating module there is code which gives object related error findings. Finally execution step it operates on the code parsed taking objects in consideration using a dedicated useful interface for it. These modules are concentrating on objects; Figure 1 show modules in object based debugging. Module second covers all operations of objects. In module three and dedicated user interface provided there is useful for developer to do operations on objects, as shown in figure 2 user operates on implemented system where user interface have important role as shown in figure. Output generator generates error messages and forward it to user interface which ultimately gives or display it to user or developer.

Factors / Capabilities	jBixbe	DBG PHP	Jswat	Backstop Tool	WPOL	CAp	CnC
Year	2006	2009	2009	2004	2006	1995	2005
Concentrated on	java	Php programs	Graphical java debugger	java	Plan object paradig m	pascal	Java
Working plattform	java	php	java	java	java	pascal	Java
Linebyline Execution	✓	✓	👎	✓	✓	✓	✓
Execution visible	👎	✓	✓	✓	👎	👎	👎
Identifies Logic error				👎		✓	✓
Web navigation available	👎	👎	👎	👎	👎	👎	👎

Common syntax logic and style errors	✓	✓	✓	✓	✓	✓	✓
Class object interaction/relationship	✓		✓	👎	👎	👎	👎
User interface friendly	✓	✓	✓	👎	👎	✓	👎
breakpoints	✓		✓	✓	✓	✓	✓
Answer questions regarding with	Application design, data structures	Php applications, data structures	Application maintanance	Except logic error	Structured knowledge representation	Common syntax, logic and style	Abstract error conditions
Popup window	👎		👎	👎	👎	👎	✓
Complicated in case	Large applications	No support new tech	Complex code	Logic errors	Complex code	Other plattforms	Derive specific error
Remark	Do not locate error	Php is old script	Require popup window	do not identify the logic error	incorporate the Plan-Object-Paradigm	Syntax logic style Errors in pascal	Produces test cases for finding error

Figure 3: Comparison Table for Various Debugging Techniques and Functionalities

RELATED WORK SHORTCOMING

Studding and analysing different literature survey following are the outcomes.

- Back in time debugging debugger have to remember history of all previous states.
- Trace oriented debugger requires more hardware power, which is practically not possible. Omniscient debugger depend on more memory because, to store history of last stages. Reverse debugging is to stop after a failure in a program has been observed and go back into the history of the execution to uncover the reason for the failure [14].
- Auto Flow can automatically reduce a large portion of irrelevant change in an early phase, eventually then locate faulty changes effectively.
- After going through literature survey came to know that developer faced some kind of problems while doing debugging.
- Major problem is that developer cannot answers about objects.
- After taking view on problems faced by developer they do not get answer to their question regarding object [14] [15].
- There is pretty need of a useful and dedicated user interface for debugging scenario [14] [15].
- Developer comfortable with using object oriented dedicated user interface for debug situations.

When complex object oriented system taken in account then traditional debuggers fails to act on object related operations and relationship between different objects. To eliminate these problems new tool should be developed on object based approach and useful dedicated user interface for it [15].

SYSTEM OVERVIEW

The source code when debug using object based debugging tool, particular object required by developer is searched and made available to developer. Developer further acting on object do the specified operation by using user interface concentrated on objects. The code file taken into proposed tool, then code parsing done for all particular objects.

After going through execution and isolates the points needed by developer needs. The parser extracted all objects from provided code file then supplied or given to execution module. This parser also converted it into intermediate forms which give response to object related errors or bugs.

As shown in Figure 4 which gives total overview for object based debugging system In code generating module there is code which gives object related error findings. In last execution step it operates on the code which is buffer and stored into array and taking objects in consideration using a dedicated useful interface for it.

IMPLEMENTATIONS

Most conventional debuggers gives only a limited level view of the given program state as well as offer little support for the exploration of large data structures, provided a system that facilitate programmers to ask the program state, helping to check object relationships in large object-oriented programs. This debugger combines multiple novel features: A new approach to debugging: Instead of exploring a single object at a time, an object based debugger facilitate the programmer to quickly extract a set of interesting objects from a potentially very large number of objects, or to check a certain property for a large number of objects.

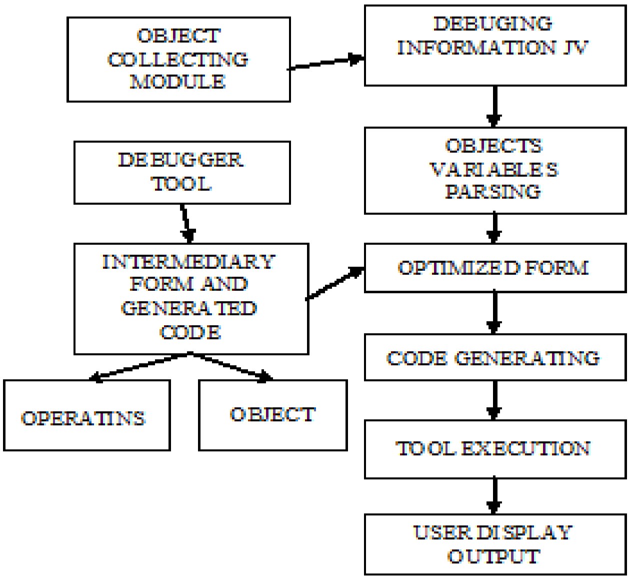


Figure 4: Overview Object Based Debugging System

A flexible tool: theoretically, evaluates its constraint expression for all members of the complex objects. This is simple to understand and to learn, yet it allows a large array of complexity of objects. Debugging easier for programmers and facilitating the development of more robust object-oriented systems.

- Perform object based debugging

- Check for the errors in code
- Check for the objects, Find errors related to objects such as variable, conditional statements, logical errors etc

These modules are concentrating on objects; Figure 5 Show modules in object based debugging. Module second covers all operations related to conditional and logical and variables of objects. In module three and dedicated user interface provided there is useful for developer to do operations on objects.

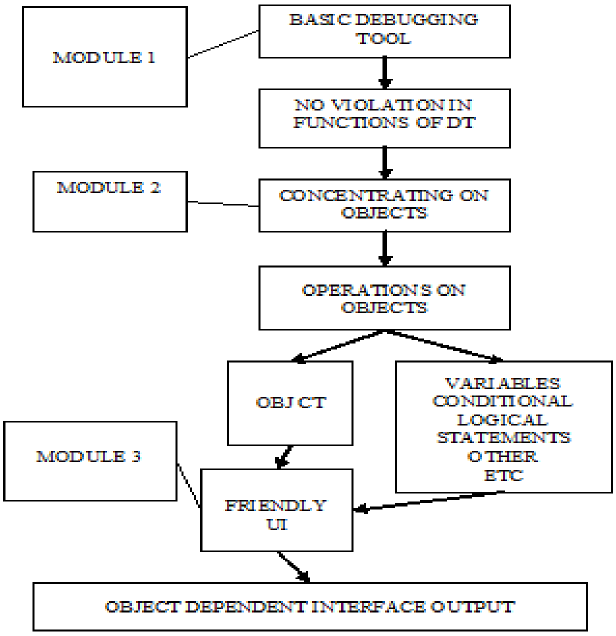

MODULE 1
BASIC DEBUGGING TOOL
NO VIOLATION IN FUNCTIONS OF DT
MODULE 2
CONCENTRATING ON OBJECTS
OPERATIONS ON OBJECTS
OBJCT
VARIABLES CONDITIONAL LOGICAL STATEMENTS OTHER ETC
MODULE 3
FRIENDLY UI
OBJECT DEPENDENT INTERFACE OUTPUT

Figure 5: System Description of Automatic Object Based Debugging

RESULTS AND DISCUSSIONS

The performance of the proposed approach is measured in terms of Recall Value and Precision Value. The performance of the proposed approach is compared with the existing tools. In the following section we discuss the performance results of the proposed approach. The detail analysis of the proposed approach is described as following. Following snapshot gives details about input file for debugging purpose to implemented system, first of all code will be line by line buffered and stored for particular line and details.

When particular object is accessed it will be shown and system will proceed further for detecting errors for which, developer is wondering in whole source code. This system has implemented on various applications and example which ultimately gives us corrected output in terms of error messages.

Figure 7 shows an example taken for case study in developed prototype implementation where it gives correct output in form of error messages.

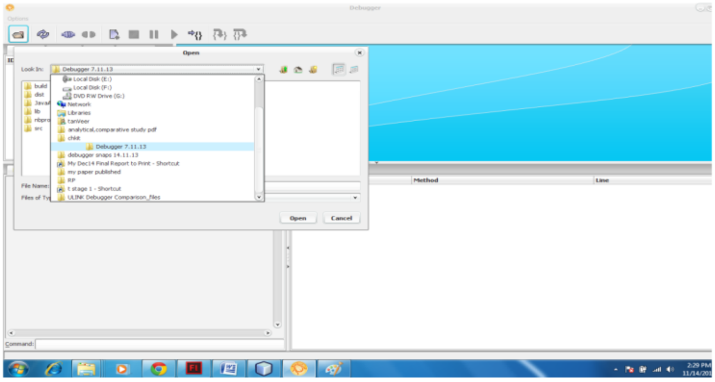
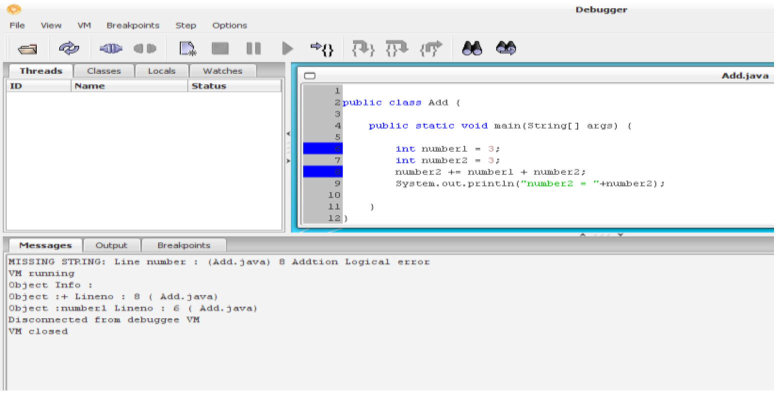


Figure 6: Example for System

CASE STUDY

In implemented prototype have considered here an example which is shown in figure 8 for case study. This designed and developed object based debugging tool system going to have its unique environment diagram as follows; as shown in figure an example is considered for debugging point of view which generates output as desired and required by user for their purpose to have defect free application.



```
 1
 2public class Add {
 3
 4    public static void main(String[] args) {
 5
 6        int number1 = 3;
 7        int number2 = 3;
 8        number2 += number1 + number2;
 9        System.out.println("number2 = "+number2);
10
11    }
12}
```

```
MISSING STRING: Line number : (Add.java) 8 Addtion Logical error
VM running
Object Info :
Object :+ Lineno : 8 ( Add.java)
Object :number1 Lineno : 6 ( Add.java)
Disconnected from debuggee VM
VM closed
```

Figure 7: Output Generated by System

CONCLUSIONS

Debugging is an important aspect in software development, where the ultimate target may be to develop code which is nil from any defect issues, Software engineers must live in the real and inadequate world where system complexity is essential thing will go to occur rapidly, Due to this The reality that debugging remains the most problematic and economical costly issue of the development cycle which indicates that engineers must work on the idea of day to day increasing complexity, prepare for the challenge from day one and acquire more as well as better debugging tools to keep increasing demands on developments. Programmers have been waiting a long time for practical automated debugging tools, they already gone a long way from near the beginning days of debugging to further advance the state of the art in this scenario, must push research towards more capable directions that take into account the way programmers actually debug in real scenarios. This proposed system provides a suitable support for the developer facilitating different features which will ultimately improve the performance of object oriented software's and their applications.

REFERENCES

1. Adrian lienhard, tudor Girba and Oscar Nierstrasz, ”Practical Object Oriented Back-In-Time Debugging”LNCS5142, pp 592-615.

2. Raimondas Lencevicius, Urs Holzle and Ambuj K. Singh, “Query-based Debugging of Object-Oriented Programs” OOPSLA 97 Atlanta, USA.

3. Mark Minas “Cyclic Debugging For pSather, a Parallel Object-Oriented Programming Language” Jan 31 2002.

4. Tanja Mayerhofer, ”Testing and Debugging UML Models Based On Fuml” ICSE 2012.

5. G. Pothier, E. Tanter, and J. Piquer, “Scalable omniscient Debugging, “Proceedings of the 22nd Annual SCM SIGPLAN Conference on Object-Oriented Programming Systems, Languages And Applications (OOPSLA’07), vol. 42, no. 10, pp.535–552, 2007.

6. C. Hofer,M. Denker, and S. Ducasse, “Design and implementation of a backward-in-time debugger,”in Proceedings of NODE’06, ser. Lecture Notes in Informatics, vol. P-88 (GI), Sep 2006, pp. 17-32.

7. J. KO and B. A. Myers, “Designing the whyline: A debugging interface for asking questions about program behavior,” in Proceedings of the 2004 conference on Human factors in computing systems. ACM Press, 2004, pp. 151–158.

8. Sai Zhang; Zhongxian Gu; Yu Lin; Jianjun Zhao “AutoFlow: An automatic debugging tool for Aspect J software” ICSM 2008. IEEE Conference on 2008, pp. 470 – 471.

9. Rossler, J. “How helpful are automated debugging tools?” User Evaluation for Software Engineering Researchers (USER), 2012 IEEE Conference Publications, pp. 13 – 16.

10. Chi-Neng Wen;shu-hsuan Chou;chih Chen ;tien-fu chen.”NUDA: A Non-Uniform Debugging Architecture and Nonintrusive Race Detection for Many core system” IEEE transaction, vol.61, 2012, pages.199-212.

11. Engblom, J “A Review of Reverse debugging” System, Software, SC and Silicon Debug Conference (S4D), 2012, pp. 1 – 6.

12. Jorge ressia, Alexandre Bergel and Oscar Nierstrasz “object centric debugging” ICSE 2012.

13. Noor Fazlida Mohd Sani, Noor Afiza Mohd Arifin and Rodziah Atan “Design of object-oriented debugger model using unified modeling language” JCSSP 2013.

14. Thakore, D. M., and Tanveer S. Beg. "AN EFFICIENT DEBUGGING TOOL FOR OBJECT ORIENTED SYSTEM." *International Journal of Computer Science* (2013).(IJCSEITR) ISSN (P): 2249-6831; ISSN(E): 2249-7943 Vol. 3, Issue 5, Dec 2013, 1-10

15. Thakore, D. M., and Tanveer S. Beg. "An Automatic Debugging Tool Extension for Object Oriented Softwares." International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013.